

Efficient implementation of RKN-type Fourier collocation methods for second-order differential equations[☆]

Bin Wang^{a,*}, Fanwei Meng^a, Yonglei Fang^b

^a*School of Mathematical Sciences, Qufu Normal University, Qufu, Shandong 273165, P.R.China*

^b*School of Mathematics and Statistics, Zaozhuang University, Zaozhuang, Shandong 277160, P.R. China*

Abstract

In this paper we discuss the efficient implementation of RKN-type Fourier collocation methods, which are used when solving second-order differential equations. The proposed implementation relies on an alternative formulation of the methods and the blended formulation. The features and effectiveness of the implementation are confirmed by the performance of the methods on two numerical tests.

Keywords: Implementation, RKN-type Fourier collocation methods, Second-order differential equations, Blended implicit methods

2010 MSC: 65L05

1. Introduction

The efficient numerical solution of implicit methods for differential equations has been the subject of many investigations in the last decades. This paper is devoted to dealing with the numerical solution of second-order differential equations, namely problems in the form

$$q''(t) = f(q(t)), \quad q(t_0) = q_0, \quad q'(t_0) = q'_0, \quad t \in [t_0, t_{\text{end}}], \quad (1)$$

where $q(t) : \mathbb{R} \rightarrow \mathbb{R}^d$ and $f(q) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is an analytic function. The solution of this system and its derivative satisfy the following variation-of-constants formula ([20]) with the stepsize h

$$\begin{aligned} q(t + \mu h) &= q(t) + \mu h q'(t) + h^2 \int_0^\mu (\mu - x) f(q(t + hx)) dx, \\ q'(t + \mu h) &= q'(t) + h \int_0^\mu f(q(t + hx)) dx. \end{aligned} \quad (2)$$

[☆]This paper was supported by National Natural Science Foundation of China (Grant No. 11401333, 11171178, 11571302), by Natural Science Foundation of Shandong Province (Grant No. ZR2014AQ003), by Foundation of Scientific Research Project of Shandong Universities (Grant No. J14LI04), and by China Postdoctoral Science Foundation (Grant No. 2015M580578).

*Corresponding author.

Email address: wangbinmaths@gmail.com (Bin Wang), fwmeng@mail.qfnu.edu.cn (Fanwei Meng), ylfangmath@163.com (Yonglei Fang)

Numerical methods of the second-order system (1) have been studied by many researchers in the last decades (see, e.g. [8, 9, 11, 12, 14, 15, 17, 18, 19, 21]), and Runge–Kutta–Nyström (RKN) methods are one of well-known methods for solving this system. In [16], the authors took advantage of shifted Legendre polynomials to obtain a local Fourier expansion of the considered system and derived a kind of collocation methods (trigonometric Fourier collocation methods). The analysis given in [16] also presents a new collocation methods (RKN-type Fourier collocation methods) for solving the second-order system (1). This kind of collocation methods is implicit and an iterative procedure is required. On the other hand, efficient implementation of implicit methods has been investigated by many researchers in recent years and we refer to [1, 2, 3, 4, 5, 6, 7] for some examples on this topic. Motivated by these publications, here we present an efficient implementation of RKN-type Fourier collocation methods, by proposing and analysing an iterative procedure based on the particular structure of the methods.

With this premise, this paper is organized as follows. Section 2 describes the derivation of RKN-type Fourier collocation methods and the structure of the discrete problem generated by the methods. In Section 3 we propose and analyze an efficient implementation of RKN-type Fourier collocation methods. Section 4 reports two numerical tests to show the features and effectiveness of the methods. The last section contains a few conclusions.

2. RKN-type Fourier collocation methods

RKN-type Fourier collocation methods are given in [16] as a by-product of trigonometric Fourier collocation methods for second-order differential equations. We now recall the derivation of the methods, and derive the most efficient formulation of the generated discrete problems.

Let us consider the restriction of problem (1) to the interval $[t_0, t_0 + h]$, with the right-hand side expanded along the shifted Legendre polynomials $\{\hat{P}_j\}_{j=0}^{\infty}$ over the interval $[0, 1]$, scaled in order to be orthonormal. Then we rewrite the function $f(q)$ in (1) as

$$f(q(t_0 + \xi h)) = \sum_{j=0}^{\infty} \hat{P}_j(\xi) \int_0^1 \hat{P}_j(\tau) f(q(t_0 + \tau h)) d\tau, \quad \xi \in [0, 1].$$

Thence the problem (1) to the interval $[t_0, t_0 + h]$ is rewritten as

$$\begin{aligned} q''(t_0 + \xi h) &= \sum_{j=0}^{\infty} \hat{P}_j(\xi) \int_0^1 \hat{P}_j(\tau) f(q(t_0 + \tau h)) d\tau, \\ q(t_0) &= q_0, \quad q'(t_0) = q'_0. \end{aligned}$$

Truncating the series at the right-hand side gives the following approximate problem

$$\begin{aligned} u''(t_0 + \xi h) &= \sum_{j=0}^{r-1} \hat{P}_j(\xi) \int_0^1 \hat{P}_j(\tau) f(u(t_0 + \tau h)) d\tau, \\ u(t_0) &= q_0, \quad u'(t_0) = q'_0. \end{aligned}$$

Introduce a quadrature formula based at k ($k \geq r$) abscissae $0 \leq c_1 \leq \dots \leq c_k \leq 1$ to deal with $\int_0^1 \hat{P}_j(\tau) f(u(t_0 + \tau h)) d\tau$. Thus we obtain an approximation of the form

$$\int_0^1 \hat{P}_j(\tau) f(u(t_0 + \tau h)) d\tau \approx \sum_{l=1}^k b_l \hat{P}_j(c_l) f(u(t_0 + c_l h)), \quad j = 0, 1, \dots, r-1,$$

where b_l with $l = 1, 2, \dots, k$ are the quadrature weights. It is natural to consider the following discrete problem as an approximation of (1)

$$v''(t_0 + \xi h) = \sum_{j=0}^{r-1} \widehat{P}_j(\xi) \sum_{l=1}^k b_l \widehat{P}_j(c_l) f(v(t_0 + c_l h)), \quad v(t_0) = q_0, \quad v'(t_0) = q'_0, \quad (3)$$

which can be solved by the variation-of-constants formula (2) in the form:

$$v_i = q_0 + c_i h q'_0 + h^2 \sum_{j=0}^{r-1} \int_0^{c_i} \widehat{P}_j(x)(c_i - x) dx \sum_{l=1}^k b_l \widehat{P}_j(c_l) f(v_l),$$

$$i = 1, 2, \dots, k,$$

where $v_i = v(t_0 + c_i h)$.

The approximation to $q(t_0 + h)$, $q'(t_0 + h)$ is then given by $q_1 = v(t_0 + h)$, $q'_1 = v'(t_0 + h)$, which can be obtained by applying the variation-of-constants formula (2) to (3) as follows:

$$q_1 = q_0 + h q'_0 + h^2 \sum_{j=0}^{r-1} \int_0^1 \widehat{P}_j(x)(1 - x) dx \sum_{l=1}^k b_l \widehat{P}_j(c_l) f(v_l),$$

$$q'_1 = q'_0 + h \int_0^1 \widehat{P}_j(x) dx \sum_{l=1}^k b_l f(v_l).$$

We compute

$$\int_0^1 \widehat{P}_j(x)(1 - x) dx = \begin{cases} \frac{1}{2}, & j = 0, \\ \frac{-1}{2\sqrt{3}}, & j = 1, \\ 0, & j \geq 2, \end{cases} \quad \int_0^1 \widehat{P}_j(x) dx = \begin{cases} 1, & j = 0, \\ 0, & j \geq 1. \end{cases}$$

Then we have the following result for $r \geq 2$:

$$q_1 = q_0 + h q'_0 + h^2 \sum_{l=1}^k (1 - c_l) b_l f(v_l),$$

$$q'_1 = q'_0 + h \sum_{l=1}^k b_l f(v_l).$$

We are now in a position to present RKN-type Fourier collocation methods for the second-order ODEs (1).

Definition 2.1. ([16]) A k -stage RKN-type Fourier collocation method for integrating the system

(1) is defined as

$$\begin{aligned}
v_i &= q_0 + c_i h q'_0 + h^2 \sum_{j=0}^{r-1} \int_0^{c_i} \hat{P}_j(x)(c_i - x) dx \sum_{l=1}^k b_l \hat{P}_j(c_l) f(v_l), \quad i = 1, 2, \dots, k, \\
q_1 &= q_0 + h q'_0 + h^2 \sum_{l=1}^k (1 - c_l) b_l f(v_l), \\
q'_1 &= q'_0 + h \sum_{l=1}^k b_l f(v_l),
\end{aligned} \tag{4}$$

where h is the stepsize, r is an integer with the requirement $2 \leq r \leq k$, (c_l, b_l) with $l = 1, 2, \dots, k$ are the node points and the quadrature weights of a quadrature formula, respectively.

Remark 1. It is remarked that the properties of the methods including the convergence, stability, and the degree of accuracy in preserving the solution, the quadratic invariant and the Hamiltonian have been studied in [16].

It is noted that the method (4) is the subclass of k -stage RKN methods with the following Butcher tableau:

$$\begin{array}{c|c} c & \bar{A} = (\bar{a}_{lm})_{k \times k} \\ \hline & \bar{b}^\top \\ & b^\top \end{array} = \begin{array}{c|c} \begin{array}{c} c_1 \\ \vdots \\ c_k \end{array} & \begin{array}{c} \bar{a}_{lm} = b_m \sum_{j=0}^{r-1} \int_0^{c_l} \hat{P}_j(x)(c_l - x) dx \hat{P}_j(c_m) \\ (1 - c_1)b_1 \quad \dots \quad (1 - c_k)b_k \\ b_1 \quad \dots \quad b_k \end{array} \\ \hline \end{array} \tag{5}$$

It is convenient to express the methods in block-matrix notation

$$\begin{aligned}
v &= u \otimes q_0 + hc \otimes q'_0 + h^2 \bar{A} \otimes I_d f(v), \\
q_1 &= q_0 + h q'_0 + h^2 (\bar{b}^\top \otimes I_d) f(v), \\
q'_1 &= q'_0 + h (b^\top \otimes I_d) f(v),
\end{aligned} \tag{6}$$

where I_d is the $d \times d$ identity matrix, \otimes is the Kronecker product, $u = (1, \dots, 1)^\top$, $v = (v_1^\top, \dots, v_k^\top)^\top$ and $f(v) = (f(v_1)^\top, \dots, f(v_k)^\top)^\top$.

It can be observed that usually the method (4) constitutes of a system of implicit equations for the determination of v_i ($i = 1, 2, \dots, k$) and it requires an iterative procedure. It is noted that usually $k \geq r$ in (4) and the iterative computation can be reduced to solve an implicit system with r equations, which is quite important for practical computations especially when $r < k$. The effective implementation of RKN-type Fourier collocation method will be discussed in next section.

3. Implementation of the methods

In this section, we propose and analyze the efficient implementation of RKN-type Fourier collocation methods.

3.1. Fundamental and silent stages

By introducing the matrices $\Omega_k = \text{diag}(b_1, b_2, \dots, b_k)$,

$$\mathcal{L}_{k,r} = \left(\int_0^{c_i} \widehat{P}_j(x)(c_i - x)dx \right)_{\substack{i=1,2,\dots,k \\ j=0,1,\dots,r-1}} \in \mathbb{R}^{k \times r}, \quad (7)$$

and

$$\mathcal{P}_{k,r} = \left(\widehat{P}_j(c_i) \right)_{\substack{i=1,2,\dots,k \\ j=0,1,\dots,r-1}} \in \mathbb{R}^{k \times r}, \quad (8)$$

the matrix \bar{A} of RKN-type Fourier collocation method given in (5) can be recast as

$$\bar{A} = \mathcal{L}_{k,r} \mathcal{P}_{k,r}^\top \Omega_k. \quad (9)$$

Theorem 3.1. *The matrix \bar{A} in (9) can be written as*

$$\bar{A} = \mathcal{P}_{k,r+2} \begin{pmatrix} X_{r,r} \\ \xi_{r-1} \xi_r e_{r-1}^\top \\ \xi_r \xi_{r+1} e_r^\top \end{pmatrix} \mathcal{P}_{k,r}^\top \Omega_k, \quad (10)$$

where

$$\xi_m = \frac{1}{2\sqrt{4m^2 - 1}},$$

$$X_{r,r} = \begin{pmatrix} \frac{1}{4} - \xi_1^2 & -\frac{\xi_1}{2} & \xi_1 \xi_2 & & & & \\ \frac{\xi_1}{2} & -\xi_1^2 - \xi_2^2 & 0 & \xi_2 \xi_3 & & & \\ \xi_1 \xi_2 & 0 & -\xi_2^2 - \xi_3^2 & 0 & \ddots & & \\ & \xi_2 \xi_3 & 0 & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & 0 & \xi_{r-2} \xi_{r-1} \\ & & & \ddots & 0 & -\xi_{r-2}^2 - \xi_{r-1}^2 & 0 \\ & & & & \xi_{r-2} \xi_{r-1} & 0 & -\xi_{r-1}^2 - \xi_r^2 \end{pmatrix}, \quad (11)$$

and e_{r-1} , e_r are the unit coordinate vectors of length $r-1$ and r , respectively.

Proof. According to [13], the shifted Legendre polynomials $\{\widehat{P}_j\}_{j=0}^\infty$ satisfy the following integration formulae

$$\begin{aligned} \int_0^x \widehat{P}_0(t)dt &= \xi_1 \widehat{P}_1(x) + \frac{1}{2} \widehat{P}_0(x), \\ \int_0^x \widehat{P}_m(t)dt &= \xi_{m+1} \widehat{P}_{m+1}(x) - \xi_m \widehat{P}_{m-1}(x), \quad m = 1, 2, \dots \end{aligned} \quad (12)$$

This yields that

$$\begin{aligned} \widehat{P}_0(x) &= \xi_1 \widehat{P}'_1(x) + \frac{1}{2} \widehat{P}'_0(x), \\ \widehat{P}_m(x) &= \xi_{m+1} \widehat{P}'_{m+1}(x) - \xi_m \widehat{P}'_{m-1}(x), \quad m = 1, 2, \dots \end{aligned}$$

We compute

$$\begin{aligned}
& \int_0^{c_i} \widehat{P}_0(x)(c_i - x)dx = c_i \int_0^{c_i} \widehat{P}_0(x)dx - \int_0^{c_i} x\widehat{P}_0(x)dx \\
& = c_i \int_0^{c_i} \widehat{P}_0(x)dx - \int_0^{c_i} x \left(\xi_1 \widehat{P}'_1(x) + \frac{1}{2} \widehat{P}'_0(x) \right) dx \\
& = c_i \int_0^{c_i} \widehat{P}_0(x)dx - \xi_1 \int_0^{c_i} x d\widehat{P}_1(x) + \frac{1}{2} \int_0^{c_i} x d\widehat{P}_0(x) \\
& = c_i \int_0^{c_i} \widehat{P}_0(x)dx - \xi_1 c_i \widehat{P}_1(c_i) + \xi_1 \int_0^{c_i} \widehat{P}_1(x)dx + \frac{1}{2} c_i \widehat{P}_0(c_i) - \frac{1}{2} \int_0^{c_i} \widehat{P}_0(x)dx,
\end{aligned}$$

and

$$\begin{aligned}
& \int_0^{c_i} \widehat{P}_j(x)(c_i - x)dx = c_i \int_0^{c_i} \widehat{P}_j(x)dx - \int_0^{c_i} x\widehat{P}_j(x)dx \\
& = c_i \int_0^{c_i} \widehat{P}_j(x)dx - \int_0^{c_i} x \left(\xi_{j+1} \widehat{P}'_{j+1}(x) - \xi_j \widehat{P}'_{j-1}(x) \right) dx \\
& = c_i \int_0^{c_i} \widehat{P}_j(x)dx - \xi_{j+1} \int_0^{c_i} x d\widehat{P}_{j+1}(x) + \xi_j \int_0^{c_i} x d\widehat{P}_{j-1}(x) \\
& = c_i \int_0^{c_i} \widehat{P}_j(x)dx - \xi_{j+1} c_i \widehat{P}_{j+1}(c_i) + \xi_{j+1} \int_0^{c_i} \widehat{P}_{j+1}(x)dx + \xi_j c_i \widehat{P}_{j-1}(c_i) \\
& \quad - \xi_j \int_0^{c_i} \widehat{P}_{j-1}(x)dx, \quad m = 1, 2, \dots
\end{aligned}$$

According to (12), we obtain

$$\begin{aligned}
& \int_0^{c_i} \widehat{P}_0(x)(c_i - x)dx = \left(\frac{1}{4} - \xi_1^2 \right) \widehat{P}_0(c_i) + \frac{\xi_1}{2} \widehat{P}_1(c_i) + \xi_1 \xi_2 \widehat{P}_2(c_i), \\
& \int_0^{c_i} \widehat{P}_1(x)(c_i - x)dx = -\frac{\xi_1}{2} \widehat{P}_0(c_i) - (\xi_1^2 + \xi_2^2) \widehat{P}_1(c_i) + \xi_2 \xi_3 \widehat{P}_2(c_i), \\
& \int_0^{c_i} \widehat{P}_j(x)(c_i - x)dx = \xi_{j-1} \xi_j \widehat{P}_{j-2}(c_i) - (\xi_j^2 + \xi_{j+1}^2) \widehat{P}_j(c_i) + \xi_{j+1} \xi_{j+2} \widehat{P}_{j+2}(c_i), \\
& \quad j = 2, 3, \dots
\end{aligned}$$

It follows from these formulae that the matrix $\mathcal{L}_{k,r}$ (7) can be rewritten as

$$\mathcal{L}_{k,r} = \mathcal{P}_{k,r+2} \widehat{X}_{r+2,r} = \mathcal{P}_{k,r+2} \begin{pmatrix} X_{r,r} \\ \xi_{r-1} \xi_r e_{r-1}^\top \\ \xi_r \xi_{r+1} e_r^\top \end{pmatrix}.$$

Then (10) holds true. ■

We have the following result about the matrix $X_{r,r}$.

Theorem 3.2. *Let $S_r = \det(X_{r,r})$. Then $\{S_r\}_{r=1}^\infty$ satisfy the recursion*

$$\begin{aligned}
S_1 &= \frac{1}{4} - \xi_1^2, \\
S_{2n} &= -\xi_{2n}^2 S_{2n-1} + \xi_1^4 \xi_3^4 \cdots \xi_{2n-1}^4, \\
S_{2n+1} &= -\xi_{2n+1}^2 S_{2n} + \frac{1}{4} \xi_2^4 \xi_4^4 \cdots \xi_{2n}^4, \quad n = 1, 2, \dots
\end{aligned}$$

Proof. The result easily follows from the Laplace expansion, by considering that, from (11), $S_1 = \frac{1}{4} - \xi_1^2$ and $S_2 = \xi_1^2 \xi_2^2 - \frac{1}{4} \xi_2^2 + \xi_1^4$. ■

From the above analysis, it follows that a k -stage RKN-type Fourier collocation method, with $k > r$, is defined by a Butcher matrix of rank r . Therefore the discrete problem can be recast in a more convenient form, whose (block) size is r , rather than k . For this purpose, let us partition the abscissae c_i , $i = 1, 2, \dots, k$ into two sets: one with r abscissae, the other with the remaining $k - r$ ones. We choose them as the first r ones and the remaining $k - r$ ones, respectively, for the sake of simplicity. According to [2, 4], the corresponding stages are called fundamental stages and silent stages, respectively. The key idea is now that the $k - r$ silent stages can be expressed as a linear combination of the r fundamental ones. To this end, let us then partition the matrices as follows:

$$\mathcal{L}_{k,r} = \begin{pmatrix} \mathcal{L}_{k,r}^{(1)} \\ \mathcal{L}_{k,r}^{(2)} \end{pmatrix}, \quad \mathcal{P}_{k,r} = \begin{pmatrix} \mathcal{P}_{k,r}^{(1)} \\ \mathcal{P}_{k,r}^{(2)} \end{pmatrix}, \quad \Omega_k = \begin{pmatrix} \Omega_k^{(1)} & \\ & \Omega_k^{(2)} \end{pmatrix},$$

where

$$\mathcal{L}_{k,r}^{(1)}, \mathcal{P}_{k,r}^{(1)}, \Omega_k^{(1)} \in \mathbb{R}^{r \times r}, \quad \mathcal{L}_{k,r}^{(2)}, \mathcal{P}_{k,r}^{(2)} \in \mathbb{R}^{(k-r) \times r}, \quad \Omega_k^{(2)} \in \mathbb{R}^{(k-r) \times (k-r)}.$$

Similarly, let us denote by $v^{(1)}$ the (block) vector, of dimension r , containing the fundamental stages, and by $v^{(2)}$ the (block) vector, of dimension $k - r$, with the silent stages. One then obtains the equations:

$$\begin{pmatrix} v^{(1)} \\ v^{(2)} \end{pmatrix} = u \otimes q_0 + hc \otimes q'_0 + h^2 \begin{pmatrix} \mathcal{L}_{k,r}^{(1)} \\ \mathcal{L}_{k,r}^{(2)} \end{pmatrix} \begin{pmatrix} \mathcal{P}_{k,r}^{(1)} \\ \mathcal{P}_{k,r}^{(2)} \end{pmatrix}^\top \begin{pmatrix} \Omega_k^{(1)} & \\ & \Omega_k^{(2)} \end{pmatrix} \otimes I_d \begin{pmatrix} f(v^{(1)}) \\ f(v^{(2)}) \end{pmatrix},$$

which can be written as

$$\begin{aligned} v^{(1)} &= u^{(1)} \otimes q_0 + hc^{(1)} \otimes q'_0 + h^2 \mathcal{L}_{k,r}^{(1)} \mathcal{P}_{k,r}^{(1)\top} \Omega_k \otimes I_d \begin{pmatrix} f(v^{(1)}) \\ f(v^{(2)}) \end{pmatrix}, \\ v^{(2)} &= u^{(2)} \otimes q_0 + hc^{(2)} \otimes q'_0 + h^2 \mathcal{L}_{k,r}^{(2)} \mathcal{P}_{k,r}^{(2)\top} \Omega_k \otimes I_d \begin{pmatrix} f(v^{(1)}) \\ f(v^{(2)}) \end{pmatrix}, \end{aligned} \tag{13}$$

where $c^{(1)} = (c_1, \dots, c_r)^\top$, $c^{(2)} = (c_{r+1}, \dots, c_k)^\top$, $u^{(1)}$ and $u^{(2)}$ are the unit vectors of length r and $k - r$, respectively.

From the first formula of (13), it follows that

$$\mathcal{P}_{k,r}^{(1)\top} \Omega_k \otimes I_d \begin{pmatrix} f(v^{(1)}) \\ f(v^{(2)}) \end{pmatrix} = (h^2 \mathcal{L}_{k,r}^{(1)})^{-1} \otimes I_d [v^{(1)} - u^{(1)} \otimes q_0 - hc^{(1)} \otimes q'_0].$$

Inserting this result into the second formula of (13) yields

$$\begin{aligned} v^{(2)} &= u^{(2)} \otimes q_0 + hc^{(2)} \otimes q'_0 + h^2 \mathcal{L}_{k,r}^{(2)} (h^2 \mathcal{L}_{k,r}^{(1)})^{-1} \otimes I_d [v^{(1)} - u^{(1)} \otimes q_0 - hc^{(1)} \otimes q'_0] \\ &= (u^{(2)} - A_1 u^{(1)}) \otimes q_0 + h(c^{(2)} - A_1 c^{(1)}) \otimes q'_0 + A_1 \otimes I_d v^{(1)}, \end{aligned} \tag{14}$$

where $A_1 = \mathcal{L}_{k,r}^{(2)} (\mathcal{L}_{k,r}^{(1)})^{-1}$. Then, by setting the matrices

$$B_1 = \mathcal{L}_{k,r}^{(1)} \mathcal{P}_{k,r}^{(1)\top} \Omega_k^{(1)}, \quad B_2 = \mathcal{L}_{k,r}^{(1)} \mathcal{P}_{k,r}^{(2)\top} \Omega_k^{(2)},$$

substituting (14) into the first formula of (13) results in a discrete problem involving only the r fundamental stages

$$\begin{aligned} v^{(1)} &= u^{(1)} \otimes q_0 + hc^{(1)} \otimes q'_0 + h^2 B_1 \otimes I_d f(v^{(1)}) + h^2 B_2 \otimes I_d f(v^{(2)}) \\ &= u^{(1)} \otimes q_0 + hc^{(1)} \otimes q'_0 + h^2 B_1 \otimes I_d f(v^{(1)}) \\ &\quad + h^2 B_2 \otimes I_d f\left((u^{(2)} - A_1 u^{(1)}) \otimes q_0 + h(c^{(2)} - A_1 c^{(1)}) \otimes q'_0 + A_1 \otimes I_d v^{(1)}\right). \end{aligned}$$

Let

$$\begin{aligned} \Psi(v^{(1)}) &= v^{(1)} - u^{(1)} \otimes q_0 - hc^{(1)} \otimes q'_0 - h^2 B_1 \otimes I_d f(v^{(1)}) \\ &\quad - h^2 B_2 \otimes I_d f\left(\hat{u} \otimes q_0 + h\hat{c} \otimes q'_0 + A_1 \otimes I_d v^{(1)}\right) \end{aligned} \quad (15)$$

with

$$\hat{u} = u^{(2)} - A_1 u^{(1)}, \quad \hat{c} = c^{(2)} - A_1 c^{(1)}.$$

The application of the simplified Newton method for solving (15) then gives

$$[I_{rd} - h^2 \tilde{C} \otimes J_0] \delta_l = -\Psi(v_l^{(1)}), \quad v_{l+1}^{(1)} = v_l^{(1)} + \delta_l, \quad (16)$$

where $J_0 = \frac{\partial f(q_0)}{\partial q}$ and

$$\tilde{C} = B_1 + B_2 A_1 = \mathcal{L}_{k,r}^{(1)} \mathcal{P}_{k,r}^{(1)\top} \Omega_k^{(1)} + \mathcal{L}_{k,r}^{(1)} \mathcal{P}_{k,r}^{(2)\top} \Omega_k^{(2)} \mathcal{L}_{k,r}^{(2)} (\mathcal{L}_{k,r}^{(1)})^{-1} \in \mathbb{R}^{r \times r}. \quad (17)$$

The following result holds true for the matrix \tilde{C} .

Theorem 3.3. *The eigenvalues of matrix \tilde{C} coincide with those of matrix $X_{r,r}$ defined in (11).*

Proof. It follows from (17) that

$$\begin{aligned} \tilde{C} &= \mathcal{L}_{k,r}^{(1)} \mathcal{P}_{k,r}^{(1)\top} \Omega_k^{(1)} + \mathcal{L}_{k,r}^{(1)} \mathcal{P}_{k,r}^{(2)\top} \Omega_k^{(2)} \mathcal{L}_{k,r}^{(2)} (\mathcal{L}_{k,r}^{(1)})^{-1} \\ &= \mathcal{L}_{k,r}^{(1)} (\mathcal{P}_{k,r}^{(1)\top} \Omega_k^{(1)} \mathcal{L}_{k,r}^{(1)} + \mathcal{P}_{k,r}^{(2)\top} \Omega_k^{(2)} \mathcal{L}_{k,r}^{(2)} (\mathcal{L}_{k,r}^{(1)})^{-1}) \\ &= \mathcal{L}_{k,r}^{(1)} (\mathcal{P}_{k,r}^\top \Omega_k \mathcal{L}_{k,r}) (\mathcal{L}_{k,r}^{(1)})^{-1} \\ &\sim \mathcal{P}_{k,r}^\top \Omega_k \mathcal{L}_{k,r} = \mathcal{P}_{k,r}^\top \Omega_k \mathcal{P}_{k,r+2} \hat{X}_{r+2,r}. \end{aligned}$$

By considering that $k > r$ and the quadrature formula (c_i, b_i) is exact for polynomials of degree no larger than $2k - 1$, we obtain:

$$\sum_{l=1}^k b_l \hat{P}_i(c_l) \hat{P}_j(c_l) = \int_0^1 \hat{P}_i(x) \hat{P}_j(x) dx = \delta_{ij}, \quad i = 0, 1, \dots, r-1, \quad j = 0, 1, \dots, r+1.$$

Thus

$$\mathcal{P}_{k,r}^\top \Omega_k \mathcal{P}_{k,r+2} = \begin{pmatrix} I_r & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

and

$$\tilde{C} \sim \begin{pmatrix} I_r & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} X_{r,r} \\ \xi_{r-1} \xi_r e_{r-1}^\top \\ \xi_r \xi_{r+1} e_r^\top \end{pmatrix} = X_{r,r}.$$

■

Remark 2. It follows from this theorem that the matrix \tilde{C} has always the same spectrum, independently of the choice of the fundamental and silent abscissae. However, the condition number of \tilde{C} is greatly affected from this choice. Clearly, a badly conditioned matrix \tilde{C} would affect the convergence of both the iterations (15) and (16). Therefore, we consider a more favorable formulation independent of the choice of the fundamental abscissae of the discrete problem in next subsection.

3.2. Alternative formulation of the discrete problem

In order to overcome the previous drawback, the basic idea is to reformulate the discrete problem by considering as unknowns the coefficients. For this purpose, let us define the (block) vectors

$$\gamma = \begin{pmatrix} \gamma_0 \\ \vdots \\ \gamma_{r-1} \end{pmatrix}, \quad \gamma_j = \sum_{l=1}^k b_l \hat{P}_j(c_l) f(v_l), \quad j = 0, 1, \dots, r-1. \quad (18)$$

Then the block vector v in (6) can be expressed as

$$v = u \otimes q_0 + hc \otimes q'_0 + h^2 \mathcal{L}_{k,r} \otimes I_d \gamma.$$

It follows from (18) that

$$\begin{aligned} \gamma &= \begin{pmatrix} \gamma_0 \\ \vdots \\ \gamma_{r-1} \end{pmatrix} = \begin{pmatrix} \sum_{l=1}^k b_l \hat{P}_0(c_l) f(v_l) \\ \vdots \\ \sum_{l=1}^k b_l \hat{P}_{r-1}(c_l) f(v_l) \end{pmatrix} \\ &= \begin{pmatrix} b_1 \hat{P}_0(c_1) & b_2 \hat{P}_0(c_2) & \cdots & b_k \hat{P}_0(c_k) \\ b_1 \hat{P}_1(c_1) & b_2 \hat{P}_1(c_2) & \cdots & b_k \hat{P}_1(c_k) \\ \vdots & \vdots & \ddots & \vdots \\ b_1 \hat{P}_{r-1}(c_1) & b_2 \hat{P}_{r-1}(c_2) & \cdots & b_k \hat{P}_{r-1}(c_k) \end{pmatrix} \otimes I_d \begin{pmatrix} f(v_1) \\ f(v_2) \\ \vdots \\ f(v_k) \end{pmatrix} \\ &= \begin{pmatrix} \hat{P}_0(c_1) & \hat{P}_0(c_2) & \cdots & \hat{P}_0(c_k) \\ \hat{P}_1(c_1) & \hat{P}_1(c_2) & \cdots & \hat{P}_1(c_k) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{P}_{r-1}(c_1) & \hat{P}_{r-1}(c_2) & \cdots & \hat{P}_{r-1}(c_k) \end{pmatrix} \begin{pmatrix} b_1 & & & \\ & b_2 & & \\ & & \ddots & \\ & & & b_k \end{pmatrix} \otimes I_d \begin{pmatrix} f(v_1) \\ f(v_2) \\ \vdots \\ f(v_k) \end{pmatrix} \\ &= \mathcal{P}_{k,r}^\top \Omega_k \otimes I_d f(v). \end{aligned}$$

Thus we obtain

$$\gamma = \mathcal{P}_{k,r}^\top \Omega_k \otimes I_d f(u \otimes q_0 + hc \otimes q'_0 + h^2 \mathcal{L}_{k,r} \otimes I_d \gamma).$$

For solving such a problem, one can still use a fixed-point iteration

$$\gamma^{m+1} = \mathcal{P}_{k,r}^\top \Omega_k \otimes I_d f(u \otimes q_0 + hc \otimes q'_0 + h^2 \mathcal{L}_{k,r} \otimes I_d \gamma^m), \quad m = 0, 1, \dots,$$

whose implementation is straightforward. One can also consider a simplified-Newton iteration. Setting

$$F(\gamma) \equiv \gamma - \mathcal{P}_{k,r}^\top \Omega_k \otimes I_d f(u \otimes q_0 + hc \otimes q'_0 + h^2 \mathcal{L}_{k,r} \otimes I_d \gamma) = 0$$

and, as before $J_0 = \frac{\partial f(q_0)}{\partial q}$, it takes the form

$$[I_{rd} - h^2 \mathcal{P}_{k,r}^\top \Omega_k \mathcal{L}_{k,r} \otimes J_0] \Delta^m = -F(\gamma^m), \quad \gamma^{m+1} = \gamma^m + \Delta^m. \quad (19)$$

We compute that

$$\mathcal{P}_{k,r}^\top \Omega_k \mathcal{L}_{k,r} = \mathcal{P}_{k,r}^\top \Omega_k \mathcal{P}_{k,r+2} \hat{X}_{r+2,r} = \begin{pmatrix} I_r & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} X_{r,r} \\ \xi_{r-1} \xi_r e_{r-1}^\top \\ \xi_r \xi_{r+1} e_r^\top \end{pmatrix} = X_{r,r}.$$

Consequently, the iteration (19) becomes

$$[I_{rd} - h^2 X_{r,r} \otimes J_0] \Delta^m = -F(\gamma^m), \quad \gamma^{m+1} = \gamma^m + \Delta^m. \quad (20)$$

3.3. Blended RKN-type Fourier collocation methods

From the arguments in the previous subsection, it can be concluded that the solution of (20) is required at each integration step when approximating the problem (1). We are going to solve such an equation by means of a blended implementation of the method. Blended implicit methods provide a general framework for the efficient solution of the discrete problems generated by block implicit methods. Many researches have been done about the study of blended implementation of numerical methods (see, e.g. [5, 6, 7]).

In [7], the authors discussed the extension of blended implicit methods for second-order problems. Following this work, we now study the blended RKN-type Fourier collocation methods for solving second-order problems (1). Following [7], we consider the following linear test equation

$$q'' = -\mu^2 q, \quad \mu \in \mathbb{R}.$$

The iteration (20) for solving this problem leads to a discrete problem in the form

$$(I_{rd} - v^2 X_{r,r} \otimes I_d) \Delta = -F(\gamma^m) \equiv \eta_1, \quad v = ih\mu \equiv ix, \quad x \in \mathbb{R}. \quad (21)$$

Then, we can define the following equivalent formulation of (21)

$$\rho^2 (X_{r,r}^{-1} \otimes I_d - v^2 I_{rd}) \Delta = \rho^2 X_{r,r}^{-1} \otimes I_d \eta_1 \equiv \eta_2 \quad (22)$$

with $\rho > 0$ a free parameter. The weighting function has the form

$$\Theta(v) = I_r \otimes (I_d - v^2 \rho^2 I_d)^{-1}$$

and the resulting blended method corresponding to (21) is formally given by

$$\Theta(v) \eta_1 + (I_{rd} - \Theta(v)) \eta_2 = [A(v) - v^2 B(v)] \Delta \equiv M(v) \Delta$$

with

$$A(v) = \Theta(v) + (I_{rd} - \Theta(v)) \rho^2 X_{r,r}^{-1} \otimes I_d, \quad B(v) = \Theta(v) X_{r,r} \otimes I_d + I_{rd} - \rho^2 \Theta(v).$$

Let

$$T(\Delta) = M(v) \Delta - \Theta(v) \eta_1 - (I_{rd} - \Theta(v)) \eta_2.$$

Consequently, it induces the blended iteration

$$\Delta_{n+1} = \Delta_n - \Theta(v) T(\Delta_n),$$

r	2	3	4	5	6	7
ρ^2	$6.455e-02$	$3.205e-02$	$1.872e-02$	$1.555e-02$	$8.465e-03$	$6.214e-03$

Table 1: A few values of ρ^2 for different r .

which can be rewritten as

$$\begin{aligned} N(v)_{\Delta_{n+1}} &= N(v)_{\Delta_n} - T(\Delta_n) \\ &= (N(v) - M(v))_{\Delta_n} + \Theta(v)\eta_1 + (I_{rd} - \Theta(v))\eta_2 \equiv (N(v) - M(v))_{\Delta_n} + \eta \end{aligned}$$

with

$$N(v) = I_r \otimes (I_d - v^2 \rho^2 I_d) \equiv \Theta(v)^{-1}, \quad \eta = \Theta(v)\eta_1 + (I_{rd} - \Theta(v))\eta_2.$$

According to the linear analysis of convergence proposed in [7], the free parameter ρ^2 is chosen as

$$\rho^2 \equiv \min\{|\lambda| : \lambda \in \sigma(X_{r,r})\},$$

which provides optimal convergence properties.

A few values of ρ^2 are listed in Table 1, for the sake of completeness.

3.4. Actual blended implementation

Let us now sketch the blended implementation of RKN-type Fourier collocation methods, when applied to a general nonlinear system. In the case of the initial value problem (1), the previous arguments can be generalized in a straightforward way, by considering that now the formulae (21)-(22) and the weighting function become

$$\begin{aligned} (I_{rd} - h^2 X_{r,r} \otimes J_0)\Delta &= -F(\gamma^m) \equiv \eta_1, \\ \rho^2 (X_{r,r}^{-1} \otimes I_d - h^2 I_r \otimes J_0)\Delta &= \rho^2 X_{r,r}^{-1} \otimes I_d \eta_1 \equiv \eta_2, \\ \theta &= I_r \otimes (I_d - \rho^2 h^2 J_0)^{-1}. \end{aligned}$$

According to the analysis stated above, the blended iteration for solving the first formula of (20) is given as follows

$$\Delta_{n+1} = \Delta_n - \theta G(\Delta_n), \quad n = 0, 1, \dots, \quad (23)$$

where

$$\begin{aligned} G(\Delta) &= \tilde{M}\Delta - \theta\eta_1 - (I_{rd} - \theta)\eta_2, \\ \tilde{M} &= \theta(I_{rd} - h^2 X_{r,r} \otimes J_0) + (I_{rd} - \theta)\rho^2 (X_{r,r}^{-1} \otimes I_d - h^2 I_r \otimes J_0). \end{aligned}$$

This iteration can be rewritten as

$$\begin{aligned} \tilde{N}_{\Delta_{n+1}} &= \tilde{N}_{\Delta_n} - G(\Delta_n) \\ &= (\tilde{N} - \tilde{M})_{\Delta_n} + \theta\eta_1 + (I_{rd} - \theta)\eta_2 \equiv (\tilde{N} - \tilde{M})_{\Delta_n} + \eta \end{aligned}$$

with

$$\tilde{N} = I_r \otimes (I_d - v^2 \rho^2 I_d) \equiv \theta^{-1}, \quad \eta = \theta\eta_1 + (I_{rd} - \theta)\eta_2.$$

From (20) and (23), we have to solve the outer-inner iteration described in Table 2. A simplified (and sometimes more efficient) procedure is given by performing exactly 1 inner iteration (i.e., that with $j = 0$ in the inner cycle in Table 2) in the above procedure and the corresponding algorithm is depicted in Table 3.

```

 $\Gamma = \mathcal{P}_{k,r}^\top \Omega_k \otimes I_d$ 
 $\Theta = h^2 \mathcal{L}_{k,r} \otimes I_d$ 
 $\Upsilon = u \otimes q_0 + hc \otimes q'_0$ 
 $\Lambda = \rho^2 X_{r,r}^{-1} \otimes I_d$ 
 $J = \rho^2 h^2 J_0$ 
 $\theta = I_r \otimes (I_d - J)^{-1}$ 
 $\tilde{M} = \theta(I_{rd} - h^2 X_{r,r} \otimes J_0) + (I_{rd} - \theta)(\Lambda - I_r \otimes J)$ 
 $\gamma^0$  given % e.g.  $\gamma^0 = \Gamma f(\Upsilon)$ 
for  $l = 0, 1, \dots$ 
   $v^l = \Upsilon + \Theta \gamma^l$ 
   $f^l = f(v^l)$ 
   $\eta_1^l = -\gamma^l + \Gamma f^l$  %  $-F(\gamma^l)$ 
   $\eta_2^l = \Lambda \eta_1^l$ 
   $\Delta^{l,0} = 0$ 
  for  $j = 0, 1, \dots$ 
     $\Delta^{l,j+1} = \Delta^{l,j} - \theta(\tilde{M} \Delta^{l,j} - \eta_2^l - \theta(\eta_1^l - \eta_2^l)),$ 
  end  $\Rightarrow$  returns  $\Delta^l$ 
   $\gamma^{l+1} = \gamma^l + \Delta^l$ 
end

```

Table 2: Outer-inner iteration for the blended implementation of RKN-type Fourier collocation methods.

```

 $\Gamma = \mathcal{P}_{k,r}^\top \Omega_k \otimes I_d$ 
 $\Theta = h^2 \mathcal{L}_{k,r} \otimes I_d$ 
 $\Upsilon = u \otimes q_0 + hc \otimes q'_0$ 
 $\Lambda = \rho^2 X_{r,r}^{-1} \otimes I_d$ 
 $J = \rho^2 h^2 J_0$ 
 $\theta = I_r \otimes (I_d - J)^{-1}$ 
 $\tilde{M} = \theta(I_{rd} - h^2 X_{r,r} \otimes J_0) + (I_{rd} - \theta)(\Lambda - I_r \otimes J)$ 
 $\gamma^0$  given % e.g.  $\gamma^0 = \Gamma f(\Upsilon)$ 
for  $l = 0, 1, \dots$ 
   $v^l = \Upsilon + \Theta \gamma^l$ 
   $f^l = f(v^l)$ 
   $\eta_1^l = -\gamma^l + \Gamma f^l$  %  $-F(\gamma^l)$ 
   $\eta_2^l = \Lambda \eta_1^l$ 
   $\Delta^l = \theta(\eta_2^l + \theta(\eta_1^l - \eta_2^l)),$ 
   $\gamma^{l+1} = \gamma^l + \Delta^l$ 
end

```

Table 3: Nonlinear iteration for the blended implementation of RKN-type Fourier collocation methods.

4. Numerical tests

In this section, a couple of numerical examples are shown to put into evidence on the features and effectiveness of the methods. As an example of the RKN-type Fourier collocation methods, we choose $k = 4$ and the following Gauss–Legendre’s quadrature

$$\begin{aligned} c_1 &= \frac{1 + \sqrt{\frac{3}{7} + \frac{2\sqrt{\frac{6}{5}}}{7}}}{2}, & c_2 &= \frac{1 + \sqrt{\frac{3}{7} - \frac{2\sqrt{\frac{6}{5}}}{7}}}{2}, \\ c_3 &= \frac{1 - \sqrt{\frac{3}{7} - \frac{2\sqrt{\frac{6}{5}}}{7}}}{2}, & c_4 &= \frac{1 - \sqrt{\frac{3}{7} + \frac{2\sqrt{\frac{6}{5}}}{7}}}{2}, \\ b_1 &= \frac{1}{2}\left(\frac{1}{2} - \frac{1}{6}\sqrt{\frac{5}{6}}\right), & b_2 &= \frac{1}{2}\left(\frac{1}{2} + \frac{1}{6}\sqrt{\frac{5}{6}}\right), \\ b_3 &= \frac{1}{2}\left(\frac{1}{2} + \frac{1}{6}\sqrt{\frac{5}{6}}\right), & b_4 &= \frac{1}{2}\left(\frac{1}{2} - \frac{1}{6}\sqrt{\frac{5}{6}}\right). \end{aligned}$$

Then $r = 2$ is chosen in (4), and the corresponding method is denoted by RKN-TFC. According to the analysis of [16], this method is of order four. In order to show the efficiency and robustness of the fourth-order method, the integrators we select for comparison are:

- RKN-TFC-B: the RKN-type Fourier collocation method RKN-TFC of order four using the blended iteration described in Table 3;
- RKN-TFC-F: the method RKN-TFC using the fixed-point iteration;
- DIRKN-F: the A-stable diagonally implicit RKN method DIRKN(2)4_{12–22} of order four in [10] using the fixed-point iteration.

In the practical computations, we set 10^{-16} as the error tolerance and 10^4 as the maximum number of each iteration.

Problem 1. Consider the following perturbed Kepler’s problem

$$\begin{aligned} q_1'' &= -\frac{q_1}{(q_1^2 + q_2^2)^{3/2}} - \frac{(2\epsilon + \epsilon^2)q_1}{(q_1^2 + q_2^2)^{5/2}}, & q_1(0) &= 1, & q_1'(0) &= 0, \\ q_2'' &= -\frac{q_2}{(q_1^2 + q_2^2)^{3/2}} - \frac{(2\epsilon + \epsilon^2)q_2}{(q_1^2 + q_2^2)^{5/2}}, & q_2(0) &= 0, & q_2'(0) &= 1 + \epsilon. \end{aligned}$$

The exact solution is

$$q_1(t) = \cos(t + \epsilon t), \quad q_2(t) = \sin(t + \epsilon t).$$

The Hamiltonian is

$$H = \frac{1}{2}(q_1'^2 + q_2'^2) - \frac{1}{\sqrt{q_1^2 + q_2^2}} - \frac{(2\epsilon + \epsilon^2)}{3(q_1^2 + q_2^2)^{3/2}}.$$

The system also has the angular momentum $L = q_1 q_2' - q_2 q_1'$ as a first integral. We take the parameter value $\epsilon = 10^{-3}$.

Methods (t, h)	CPU time	Iterations	Solution error	Hamiltonian error	Angular momentum error
RKN-FCM-B (50, 0.4)	0.029	1423	-2.149	-9.248	-9.069
RKN-FCM-F (50, 0.4)	0.709	71438	0.300	-3.143	-4.126
DIRKN-F (50, 0.4)	0.132	9676	-0.544	-0.627	-0.753
RKN-FCM-B (50, 0.2)	0.064	3028	-3.354	-11.700	-11.524
RKN-FCM-F (50, 0.2)	0.531	52204	-0.196	-4.930	-5.558
DIRKN-F (50, 0.2)	0.111	5966	-1.202	-3.056	-3.057
RKN-FCM-B (50, 0.1)	0.080	3285	-4.558	-14.002	-13.875
RKN-FCM-F (50, 0.1)	0.257	23490	-0.711	-6.432	-7.038
DIRKN-F (50, 0.1)	0.085	5486	-3.006	-4.887	-4.88
RKN-FCM-B (100, 0.4)	0.090	3841	-1.879	-8.658	-8.479
RKN-FCM-F (100, 0.4)	1.692	162856	-0.255	-2.785	-3.822
DIRKN-F (100, 0.4)	0.365	26334	-0.297	-0.627	-0.753
RKN-FCM-B (100, 0.2)	0.137	7048	-3.085	-11.109	-10.932
RKN-FCM-F (100, 0.2)	0.866	84425	0.129	-4.413	-5.255
DIRKN-F (100, 0.2)	0.134	9952	-0.603	-2.750	-2.752
RKN-FCM-B (100, 0.1)	0.177	7573	-4.289	-13.461	-13.331
RKN-FCM-F (100, 0.1)	0.529	46986	-0.484	-5.900	-6.736
DIRKN-F (100, 0.1)	0.164	9981	-2.441	-4.586	-4.587

Table 4: Results for Problem 1.

We solve the problem in the intervals $[0, 50]$ and $[0, 100]$ with different stepsizes $h = 0.4, 0.2, 0.1$. Table 4 lists the CPU time, the total numbers of iterations, the solution errors, Hamiltonian errors and angular momentum errors. It is noted here that all the errors are given by the logarithm of the corresponding results.

Problem 2. The Hénon-Heiles Model is created for describing stellar motion and it can be expressed by the following form

$$\begin{aligned}
q_1''(t) &= -q_1(t) - 2q_1(t)q_2(t), \quad q_1(0) = \sqrt{\frac{11}{96}}, \quad q_1'(0) = 0, \\
q_2''(t) &= -q_2(t) - q_1^2(t) + q_2^2(t), \quad q_2(0) = 0, \quad q_2'(0) = \frac{1}{4}.
\end{aligned}$$

The Hamiltonian function of the system is given by

$$H(p, q) = \frac{1}{2}(q_1'^2 + q_2'^2) + \frac{1}{2}(q_1^2 + q_2^2) + q_1^2 q_2 - \frac{1}{3}q_2^3.$$

We solve the problem in the intervals $[0, 50]$ and $[0, 100]$ with different stepsizes $h = 0.4, 0.2, 0.1$. See Table 5 for the CPU time, the total numbers of iterations, the solution errors and Hamiltonian errors.

From the numerical tests, one can then conclude that the proposed blended implementation of RKN-type Fourier collocation methods turns out to be more robust and efficient than the same method and the RKN method using the fixed-point iteration.

Methods (t, h)	CPU time	Iterations	Solution error	Hamiltonian error
RKN-FCM-B (50, 0.1)	0.063	2989	-5.806	-8.915
RKN-FCM-F (50, 0.1)	0.048	3004	-2.145	-5.170
DIRKN-F (50, 0.1)	0.058	3979	-5.545	-5.547
RKN-FCM-B (50, 0.05)	0.113	4996	-7.010	-10.121
RKN-FCM-F (50, 0.05)	0.085	5000	-2.754	-6.005
DIRKN-F (50, 0.05)	0.092	6187	-6.653	-7.054
RKN-FCM-B (50, 0.025)	0.198	8012	-8.214	-11.325
RKN-FCM-F (50, 0.025)	0.158	8906	-3.359	-6.711
DIRKN-F (50, 0.025)	0.174	10018	-7.828	-8.560
RKN-FCM-B (100, 0.1)	0.108	5981	-5.301	-7.900
RKN-FCM-F (100, 0.1)	0.080	6004	-1.654	-4.338
DIRKN-F (100, 0.1)	0.091	7958	-4.247	-5.245
RKN-FCM-B (100, 0.05)	0.206	9996	-6.504	-9.105
RKN-FCM-F (100, 0.05)	0.162	10000	-2.253	-4.927
DIRKN-F (100, 0.05)	0.167	12366	-5.747	-6.750
RKN-FCM-B (100, 0.025)	0.368	16025	-7.708	-10.309
RKN-FCM-F (100, 0.025)	0.300	17801	-2.854	-5.528
DIRKN-F (100, 0.025)	0.318	20035	-7.193	-8.253

Table 5: Results for Problem 2.

5. Conclusions

In this paper we propose and analyze an efficient iterative procedure for solving the second-order differential equations (1) generated by the application of RKN-type Fourier collocation methods. The proposed implementation turns out to be robust and efficient. Two numerical tests confirm the effectiveness of the proposed iteration when numerically solving second-order differential equations.

Last but not least, it is noted that there are still some issues which will be further considered.

- A new splitting procedure has been developed recently for the implementation of implicit methods and we refer the reader to [1, 3] for example. We will consider the novel technique for RKN-type Fourier collocation methods in a future research.
- Trigonometric Fourier collocation methods (TFCMs) have been formulated in [16] for solving $q'' + Mq = f(q)$ with a matrix M and the coefficients of the methods depend on matrix-valued functions. Therefore, the implementation of these methods would be different and complicated. Another issue for future exploration is the research of efficient implementation for the TFCMs.
- The shifted Legendre polynomials are chosen in [16] as an orthonormal basis to give the formulation of the methods. It is noted that a different choice of the orthonormal basis would modify the scheme of the methods as well as the analysis of the implementation, which will be considered in future investigations.

References

- [1] L. Brugnano, G.F. Caccia, F. Iavernaro, Efficient implementation of Gauss collocation and Hamiltonian boundary value methods, *Numer. Algo.* 65 (2014) 633–650.
- [2] L. Brugnano, G.F. Caccia, F. Iavernaro, Hamiltonian Boundary Value Methods (HBVMs) and their efficient implementation, *mathematics in engineering, science and aerospace mesa*, 5 (2014) 343–411.
- [3] L. Brugnano, F. Iavernaro, C. Magherini, Efficient implementation of Radau collocation methods, *Appl. Numer. Math.* 42 (2015) 100–113.
- [4] L. Brugnano, F. Iavernaro, D. Trigiante, A note on the efficient implementation of Hamiltonian BVMs, *J. Comput. Appl. Math.* 236 (2011) 375–383.
- [5] L. Brugnano, C. Magherini, Blended implementation of block implicit methods for ODEs, *Appl. Numer. Math.* 42 (2002) 29–45.
- [6] L. Brugnano, C. Magherini, The BiM code for the numerical solution of ODEs, *J. Comput. Appl. Math.* 164–165 (2004) 145–158.
- [7] L. Brugnano, C. Magherini, Blended implicit methods for solving ODE and DAE problems, and their extension for second-order problems, *J. Comput. Appl. Math.* 205 (2007) 777–790.
- [8] J.C. Butcher, *Numerical Methods for Ordinary Differential Equations*, Second Edition, J. Wiley, Chichester, 2008.
- [9] J.M. Franco, Runge-Kutta-Nyström methods adapted to the numerical integration of perturbed oscillators, *Comput. Phys. Comm.* 147 (2002) 770–787.
- [10] J.M. Franco, I. Gómez, Accuracy and linear stability of RKN methods for solving second-order stiff problems, *Appl. Numer. Math.* 59 (2009) 959–975.
- [11] B. Garcia-Archilla, J.M. Sanz-Serna, R.D. Skeel, Long-time-step methods for oscillatory differential equations, *SIAM J. Sci. Comput.* 20 (1998) 930–963.
- [12] E. Hairer, C. Lubich, G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed., Springer-Verlag, Berlin, Heidelberg, 2006.
- [13] E. Hairer, S.P. Nørsett, G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer-Verlag, Berlin, second edition, 1996.
- [14] A.A. Kostı, Z.A. Anastassi, T.E. Simos, An optimized explicit Runge-Kutta-Nyström method for the numerical solution of orbital and related periodical initial value problems, *Comput. Phys. Comm.* 183 (2012) 470–479.
- [15] T.E. Simos, Exponentially-fitted Runge-Kutta-Nyström method for the numerical solution of initial-value problems with oscillating solutions, *Appl. Math. Lett.* 15 (2002) 217–225.
- [16] B. Wang, A. Iserles, X. Wu, Arbitrary-order trigonometric Fourier collocation methods for multi-frequency oscillatory systems, *Found. Comput. Math.* 16 (2016) 151–181.
- [17] B. Wang, G. Li, Bounds on asymptotic-numerical solvers for ordinary differential equations with extrinsic oscillation, *Appl. Math. Modell.* 39 (2015) 2528–2538.
- [18] B. Wang, X. Wu, J. Xia, Error bounds for explicit ERKN integrators for systems of multi-frequency oscillatory second-order differential equations, *Appl. Numer. Math.* 74 (2013) 17–34.
- [19] X. Wu, B. Wang, J. Xia, Explicit symplectic multidimensional exponential fitting modified Runge-Kutta-Nyström methods, *BIT* 52 (2012) 773–795.
- [20] X. Wu, X. You, B. Wang, *Structure-Preserving Algorithms for Oscillatory Differential Equations*, Springer-Verlag, Berlin, Heidelberg, 2013.
- [21] A. Xiao, Y. Tang, Order properties of symplectic Runge-Kutta-Nyström methods, *Comput. Math. Appl.* 47 (2004) 569–582.